In: Proc. PAAM ’96, London, U.K., April 22-24, 1996, pp. 45–59

Constraint-based Information Gathering for a Network Publication System

Uwe M. Borghoff and Remo Pareschi
Rank Xerox Research Centre, Grenoble Laboratory
6, chemin de Maupertuis. F-38240 Meylan, France
Tel.: +33 7661-5076. Fax: +33 7661-5099
E-mail: {borghoff,pareschi}@grenoble.rxrc.xerox.com

Harald Karch
Rank Xerox Germany Systems Operations
Werftstraße 37. D-40549 Düsseldorf (Heerdt), Germany
Tel.: +49 511 6269-38. Fax: +49 211 990-7570
E-mail: karch@braunschweig.netsurf.de

Martina Nöhmeier and Johann H. Schlichter*
Institut für Informatik
Technische Universität München. D-80290 München, Germany
Tel.: +49 89 4505-5220. Fax: +49 89 4505-5222
E-mail: {noehmeie,schlicht}@informatik.tu-muenchen.de

Abstract

The Internet and the World-Wide Web (WWW) are revolutionizing knowledge exchange by linking heterogeneous information repositories into a kind of gigantic world-wide digital library. Yet up until now, knowledge management on the WWW has mainly been provided by navigation tools like Mosaic and Netscape, and by engines like Alta Vista, Lycos and Yahoo which support navigation by automating the search for user-relevant WWW sites. The simplicity of this paradigm has been the key to the initial success of the Web infrastructure but now falls short of more complex applications needed by an ever-growing community of users. Prominent among these needs is flexible information gathering from multiple knowledge sources to ad-hocratically serve the requests of specific user groups. For instance, Network Publication Systems (NPS) for large organizations need flexible integration of enquiry information like Who’s Who services and tables of contents of journals with E-print archival material, as well as flexible adaptation of local query services. Agent technology can provide the right answer to these demands. In this paper, we describe agent-based information gathering on the WWW in the context of a NPS for the European Physicist Society. In our approach, we exploit *constraints* to implement information gathering with maximal flexibility.

Key words. Internet, WWW, network publication systems, information gathering, constraints.

---

[*] Work performed while visiting the Rank Xerox Research Centre in Grenoble

1 Introduction

In spite of the enormous amount of information available, the World-Wide Web has so far been accessible essentially through simple navigation tools like Mosaic and Netscape. The simplicity of this paradigm has been the key to the initial success of the Web infrastructure but now falls short of more complex applications needed by an ever-growing community of users. Prominent among these needs is flexible information gathering from multiple knowledge sources to ad-hocratically serve the requests of specific user groups. Agent technology is an obvious candidate to fulfill this demand. Through the use of remote programming tools like Java and Telescript, autonomous software agents can access heterogeneous information repositories to select and merge appropriate knowledge to satisfy user requests. In doing so, they can leverage basic yet powerful indexing facilities such as Lycos, Alta Vista, Yahoo and other retrieval engines.

However, aside from such infrastructural support, the issue on how to design and implement agents of this kind remains open. In this paper, we show how *constraints*, a long-known construct from artificial intelligence and computer science, can give the right answer. Constraints have in the past been exploited essentially for the combinatorial optimization of computationally hard problems. These capabilities have been embedded into full-blown programming environments, either of the object-oriented (Freeman-Benson 1990) or of the logic programming (Hentenryck and Saraswat 1995) breed. More recently, constraints have been used to capture partial information in a world of concurrent communicating agents (Henz *et al.* 1995, Saraswat 1989). The possibility of exploiting such a view of concurrency in the context of distributed knowledge management was made explicit by Andreoli *et al.* (1994–1996). For the practical purpose of knowledge management on the Web, the main advantages of using constraints can be summarized as follows:

1. from the users point of view, constraints can be used to flexibly specify *partial requests*, namely requests which may leave underspecified, certain aspects of the requested information.
2. from the point of view of the agent platform, constraints can be used to create *dependencies* among concurrent subrequests into which an initial request is decomposed. Constraints also provide *concurrency control* among the agents managing the queries.
3. for information providers, constraints can be used to dynamically augment local query interfaces by *filtering* result items on a per constraint basis.

There is a large variety and number of multiagent applications for knowledge management on the Web where these capabilities can be exploited: among others, bargain finding, dynamic assemblage of virtual catalogs, data warehousing from backend repositories, agent-based document construction and customization can all be supported through this paradigm. In this paper, we describe a case from the domain of network publication systems (NPS), specifically a system embedded in a European distributed document database for physics which will be extended to other learned fields like mathematics, computer science and chemistry. This project is being developed collaborative between Rank Xerox Germany System Operations, the Grenoble Lab of the Rank Xerox Research Centre and several university departments.

The paper is organized as follows. After a brief discussion of related work in Sect. 2, we present in Sect. 3 the application environment, namely the Physicists Network Publishing System (PNPS). PNPS serves as a testbed for the knowledge broker framework, our agent-based approach for information gathering using constraints. Sect. 4 describes, in detail, the architectural framework and illustrates the four basic components, user interface, broker hierarchy, wrappers, and external archives. Sect. 5 concludes the paper.

2 Related Work

Well-established publishing systems, like Gopher and the World-Wide Web, provide a seamless information space in the Internet, at least as far as graphical browsing is concerned. Index and search subsystems appeared hand in hand with the rapid growth in the amount of information and in the number of users having specific needs. Obraczka *et al.* (1993) and Schwartz *et al.* (1992) give an overview of resource discovery approaches.

One of the earliest Internet indexing approaches were the Wide-Area Information Servers (WAIS) (Kahle and Medlar 1991), providing a Z39.50-based search and retrieval interface, and Archie (Emtage and Deutsch 1992). Archie periodically contacts a set of registered servers to gather a file index. Similar to that, Aliweb contains user-written summaries of server contents that are displayed on request.

More recently, with Glimpse (GLobal IMPlicit SEarch) (Manber and Wu 1994) an index/search subsystem has been installed that allows sophisticated searches over entire file systems. Among others, it allows misspelling and regular expression searches over non-uniform information including many types of documents. At the University of Karlsruhe, a prominent application has been realized on top of Glimpse, namely the sophisticated search facility for a large collection of computer science bibliographies.

Although multi-source index/search subsystems have already been built for Gopher, with Veronica, and for WWW, with Alta Vista, Lycos, and the World-Wide Web Worm (WWWW), retrieval engines or retrieval support systems for heterogeneous information are still open research fields (Barbara 1993). Early prototypes have however got an airing. The system Inquery, currently being developed at Amherst University by Callan *et al.* (1992, 1995), calculates the appropriateness of heterogeneous information sources with respect to a given query. It chooses the best fitting sources and conducts the search processes. At Stanford University gGLOSS (generalized Glossary-Of-Servers Server) addresses a similar idea. Following Gravano and Garcia-Molina (1995), gGLOSS keeps sophisticated statistics on available databases to determine an estimate of which databases are most appropriate for a given query. The search process is performed in a ranked list of databases. In contrast to Archie, which gathers an index without having a particular query in mind, Inquery and gGLOSS provide their indexes dynamically and are tailored to individual needs, viz a single query. The indexes then guide individual searches across the set of servers.

As soon as appropriate index/search prototypes were implemented, intelligent agents (CACM 1994, Wooldridge and Jennings 1995) or knowledge brokers (Barbara and Clifton 1992) started to exploit these subsystems. Harvest (Bowman *et al.* 1994a), for example, exploits as an index/search subsystem, both Glimpse and Nebula (Bowman *et al.* 1994b). Knowledge brokers are autonomous entities that may collaborate, negotiate, and coordinate, but which by no means can be coerced into activities such as searching information or answering a query whose scope does not conform with the broker's ability in query handling (Andreoli *et al.* 1995). Thus, knowledge brokers are generally used in combination with index/search subsystems.

In the Constraint-Based Knowledge Broker model (CBKB), constraints have been introduced to flexibly manage the search space of broker agents, as well as to flexibly adapt user requests and answers from information providers. Andreoli *et al.* (1996) present the theoretical background of CBKB. Protocol issues within CBKB are addressed by Arcelli *et al.* (1995) and Borghoff *et al.* (to appear). Fikes *et al.* (1995) also use logic-based models to capture the domain of expertise of information brokers. Rather than using constraints, their modeling language is based on a predicate logic with contexts. The Tsimmis project (Chawathe *et al.* 1994) takes a different approach using a self-describing object model for the internal representation of information and queries.

It should be pointed out, however, that our approach differs from other frameworks for agent-based information gathering on the Internet not only in the technology, but also in the assumptions we make with respect to the development of a "cyber-economy." Indeed, we differ from those approaches which view the Internet as a kind of global market where agents – roaming over open electronic domains – will meet and gather information, possibly leading to business transactions. On the contrary, we see the Internet as evolving into a galaxy of *intranets*, linking together information providers and users around common interests. On the basis of these social and economic considerations, technological choices can be consequently specialized to optimally fit the requirements of specific intranets and user communities. Tools like Java, that provide capabilities for easy customizations of both client and server sides, appear particularly well-suited for this purpose.

This paper documents one such case of specialization, namely the adaptation of an agent infrastructure for constraint-based information gathering to the requirements of a network publication system for research and education.

3 Physicists Network Publishing System

In this section we describe the application environment, namely the Physicists Network Publishing System (PNPS), that serves as a testbed for the knowledge broker framework.

PNPS is embedded in a European distributed document database for physics (DDD-Physics). The DDD-Physics is a coordinated effort to organize, and to some degree standardize, the document-servers of physics departments and related research institutions and combine them with services of commercial providers such as publishers, database hosts, or libraries using common search interfaces. It will allow searches in a somewhat unified way over all diverse distributed document servers. This effort will be extended into other learned fields in Germany like mathematics, computer science and chemistry.

The PNPS will serve the remote user from an html browser to order a document from several document databases for printing-on-demand on local commercial copy centers.

3.1 Basic Architecture of the PNPS

There are three major functional subsystems:

1. *archive management.* There may be different archive systems, each managed locally by their providers (e.g. commercial publishers, scientific archives like E-print servers LANL, SISSA, or local department servers). The archives may be based on different management software and are integrated by the knowledge brokering system. This aspect of the subsystem will be discussed in detail in Sect. 4.
2. *production management and controlling.* The production management subsystem handles incoming jobs from network clients. A job describes a workflow, which in the simplest case is a printing task of a network document.

3. *clients and communication infrastructure.* The complete production process must be managed including accounting, authorization, billing, and logistics information.

3.2 Production Management and Controlling

Rank Xerox Germany System Operations (RXG-SO) has developed a printing-on-demand (POD) system based on Xerox printer technology in a local area network.

Special focus was placed on implementing the specific functions and requirements of the customers, and designing a highly generic rescalable system. In contrast to *network printing* application classes, where the emphasis is to have a highly universal network print server with strongly varying print jobs, the POD-system addresses an application class which may be called *archive printing*. The characteristics of archive printing are quite similar to the design model of local POD-systems:

- the system works in a nearly static and well-defined production environment. This means that the formats of the printable data are specified in advance, since they are to be stored in local archives.
- the system is interfaced to external control systems, for example, jobs are input from external systems, and information (accounting, document-keys, etc.) has to be returned to these external systems.
- the local archives are managed by system administrators, and there is a validation process for all new documents.
- documents are described by both page description languages (PDL) and in raster format.
- the print jobs are mixed with logistic information (delivery sheets, DP-data with forms, etc.) and archive documents.
- the print jobs may be huge (100.000 pages) requiring a focus on crash recovery.
- jobs enter the system anonymously, where they have to be identified, classified, and controlled.
- all printers serve as one printer pool, which can be controlled centrally.

3.3 Clients and Communication Infrastructure

Network clients use a general purpose interface (e.g. WWW) to input jobs into the production system. The communication infrastructure provides functions for authentication, authorization and accounting of services. Today, the first version of an accounting service exists. It is installed at the University of Oldenburg, to authenticate client orders by password.

Billing servers use the accounting information and other user related data (e.g. rating) to produce the invoices for the printing-on-demand service. The status of a job may be reviewed by the user client after it has passed the production stage.

3.4 Current Status and Futures

The dedicated POD-system was implemented for commercial applications in a LAN, following the operational model given above. In a second step it was extended to distributed clients and archives in the Internet in a joint venture with the Physics Department of the University of Oldenburg, Germany. The prototype systems have proven to work in production environments.

The next steps include: opening the system to WAN solutions with multiple distributed archive and production sites, use of information brokering services for global integration of local sites, supporting individual profiles and interfaces, and the full integration of authentication, authorization, and accounting services. Furthermore, the setup of communication infrastructure services adapted to high-speed transport and commercial applications is crucial to the success of the entire system.

In the following, the use of an information brokering service for the global integration of local sites with individual profiles and interfaces will be addressed in more detail.

4 Archive Management through Knowledge Brokers

In this section we motivate and discuss the architectural aspects of the agent-based archive management system giving a brief description of the major components and their functionality including: (i) the interface which mediates between the user and the knowledge broker agents, (ii) the internal information representation used by the knowledge brokers, (iii) the partially archive-dependent back-end encapsulating the information which is necessary to integrate external data repositories.

4.1 Overview

As mentioned in Sect. 3.1, there already exist numerous documents in physics and other fields being archived in distributed heterogeneous information repositories which can be accessed via specific search subsystems. Unfortunately, these repositories often differ in many respects:

- first of all, both the structure and the significance of the information provided by the index may highly differ.
- as a consequence, the kind of search which can be performed ranges from full text search, resulting in lines of documents matched by the query, to requests on the basis of structured meta-information only.
- furthermore, the means offered to specify requests range from formal languages based on first order logic – which often allow flexible combinations of typed or untyped keywords but which force the user to acquire detailed knowledge about their syntax – to a static number of preselected, sometimes even untyped keywords, which suffer from limiting the extent to which more complex queries can be built.
- additionally, the structure as well as the significance of the results returned by the search engine is far from being homogeneous, depending on which method was used to create the index.
- last but not least, there are only few possibilities offered allowing the user to customize the applied query/answer format to their own needs in order to overcome problems arising from the heterogeneity of data repositories.

This variety of information formats, access interfaces and protocols causes many difficulties if a user wants to search for information stored over different archives. Indeed, a main problem is specifying related requests over different data repositories. Furthermore, it is generally impossible for a user to compare or combine results obtained from parallel or subsequent searches because of differences in attribute schemata. Finally, this heterogeneity is also one important obstacle that prevents the support of complex queries which allow transitive request combinations, breaking down the barriers between different archives. As a consequence, information stored in distributed, heterogeneous archives requiring the use of different search tools can only be retrieved in a fragmented way. This fragmentation originates from a lack of means to express *interdependencies* between search results and to specify the *context* for a request in terms of the results of precedent requests (Grötschel and Lügger 1995, Sanchez 1994).

Constraint Based Knowledge Brokers (CBKBs) have been exploited to overcome the problem of heterogeneity of data representations, and enable the user to search for information in a wide range of data repositories, including different kinds of indexes and search tools, in a less fragmentary way.

4.2 Architecture

Fig. 1 shows the current architecture of the CBKB system for the PNPS application. It consists of four major components:

1. the user interface that mediates between the user and the constraint-based knowledge brokers. Currently, the interface consists of dialogue-based components for permanent request-session control, iterative request and subrequest specification, and combination as well as representation of results.
2. the knowledge brokers which handle constrained requests and corresponding answers.
3. the wrappers that consist of an independent data repository part, which mediates between the constraint-based knowledge brokers and several database-dependent components, implementing the various interfaces to the available external data repositories. These interfaces encapsulate all the information which is specific to the corresponding data repository; refer also to the paper by Borghoff and Schlichter (1995) for more details on the mediating aspects of the architecture.
4. the external information repositories which are accessible via the corresponding interface using a variety of supported protocols, e.g. http.

Using this broker system to exploit a wide range of physics-specific archives, many of the above mentioned shortcomings can be overcome by flexibly adding missing functionalities to the individual data access interfaces.

First of all, it is possible to homogenize or even extend the admissible types of requests with respect to the different data repositories to be accessed, even when the accessed data repositories do not exactly match the type of request specification.

Beside the well-known attribute/value pairs, the request specification includes logical operators $=$, $\neq$, $<$, etc., and information thresholds, used for specifying concurrency control among related requests (e.g. whether results have to be returned from one query to start another one). Furthermore, results from different subrequests can be combined to obtain the desired result. This feature enables the user to perform quite more powerful searches than with standard search tools.

Add to this the fact that the number and kind of data repositories that may be accessed via the knowledge brokers is easily extensible. This is because the integration of additional

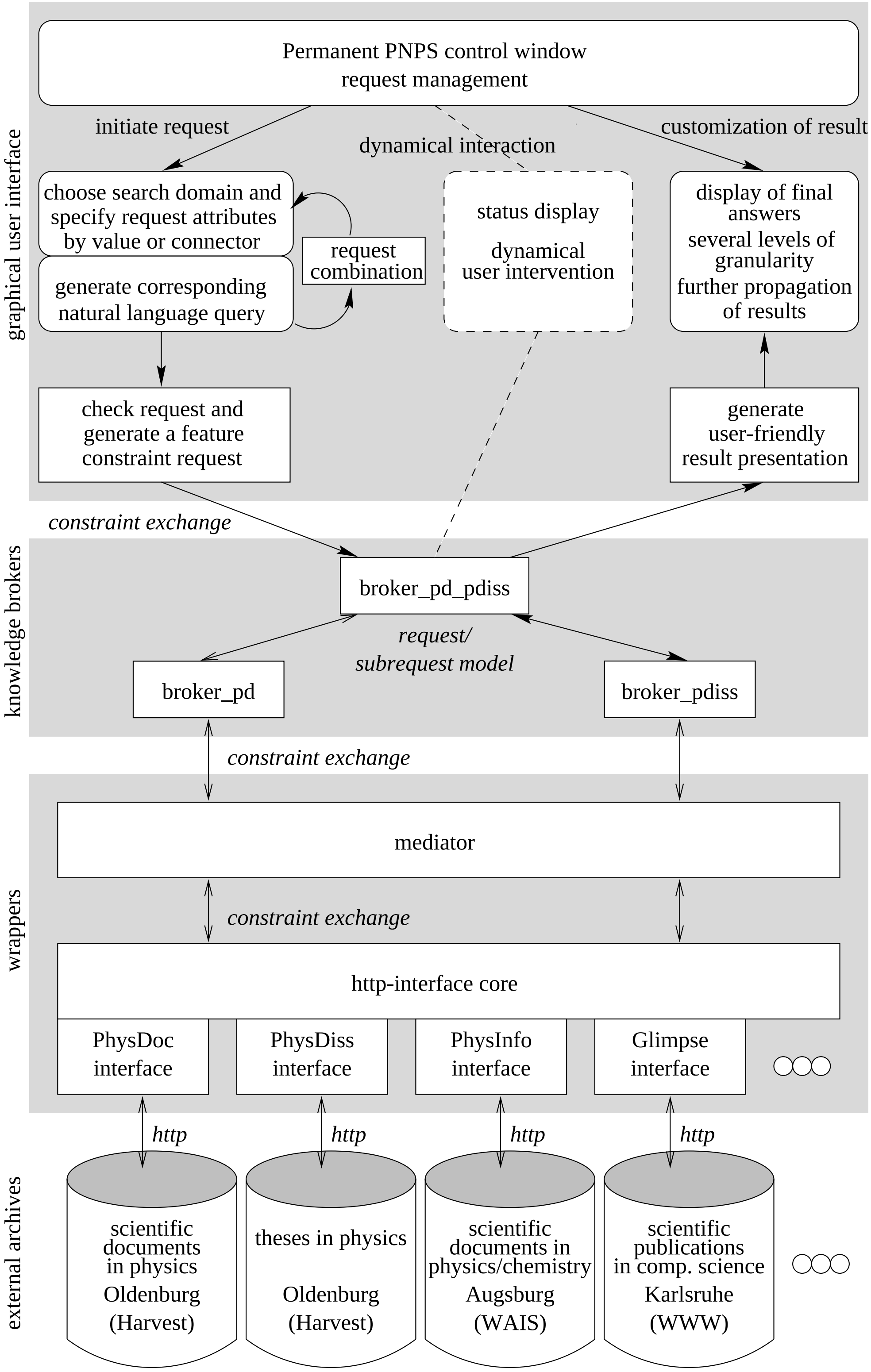


Figure 1: Architecture

repositories is a straightforward process affecting mostly just the data-dependent part of the wrapper.

Finally, an interactive user interface allows the user to dynamically widen or narrow the scope of the search during execution. This may be extremely useful when it becomes evident that the request as initially specified is not leading to a satisfying result.

In the following sections the individual components of the knowledge broker system are described in more detail, with a discussion of the requirements which they have to fulfill, of their functionality and of the current status of implementation.

4.3 User Interface

This section describes some problems and guidelines which had to be considered for the design of the user interface as well as the approach that was taken to fulfill these requirements.

4.3.1 Requirements

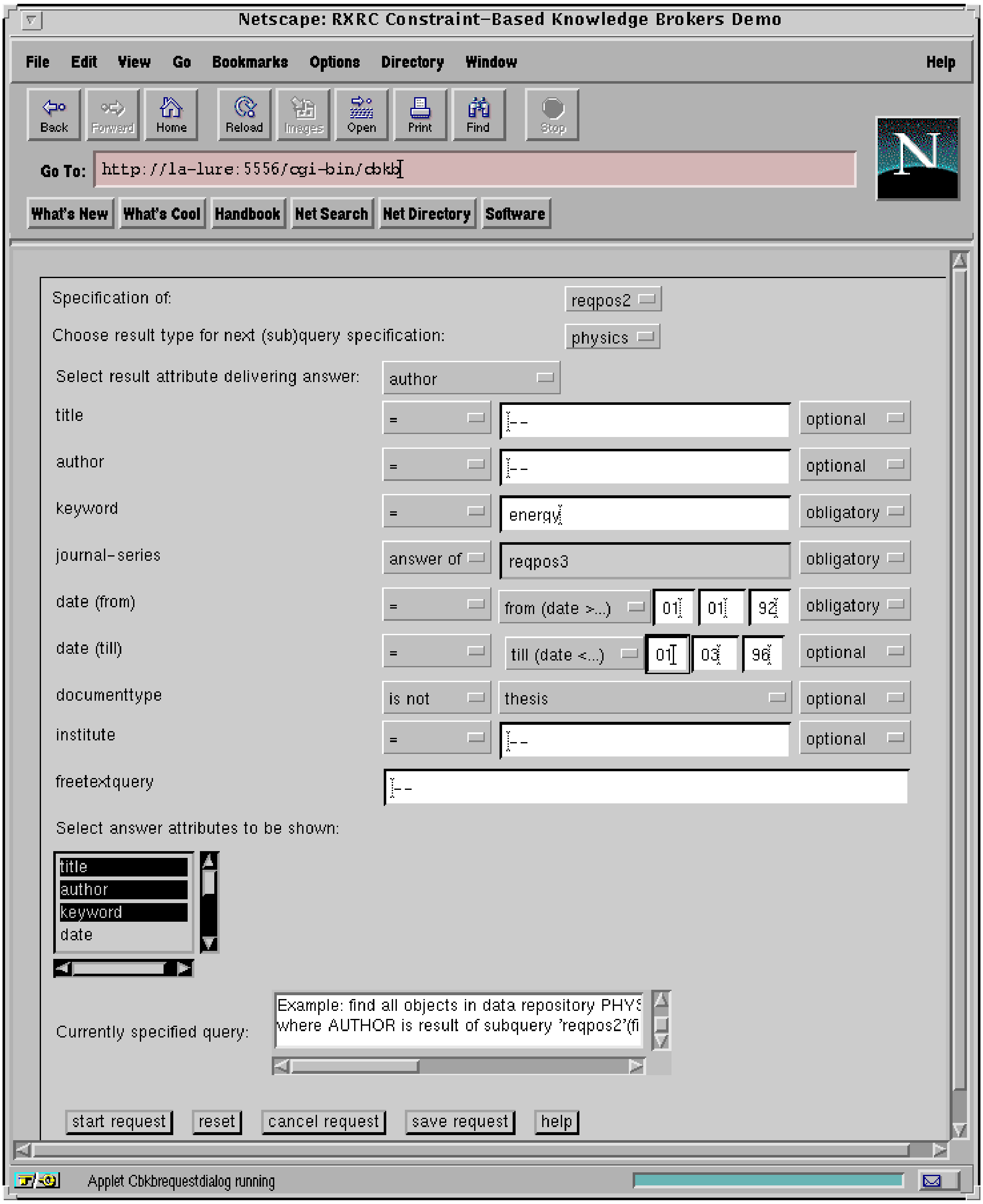
To support users, we have designed a graphical user interface which translates the content-based, domain-oriented natural-language world of the physicists accessing the system and the formal, constraint-based representation of knowledge used by the broker agents (Andreoli *et al.* 1995).

- first of all, we have to take into consideration that the knowledge broker system should comply with the needs of both the experienced information retrieval expert and the physicist who often is not used to the internal knowledge representation of the broker system and therefore might prefer to specify requests in a more content-based manner.
- one of the major tasks the user front-end has to fulfill is to allow access to the PNPS brokering services from the working environment preferred by the user, without forcing them to get used to unknown applications, to use a specific type of hardware or to install lots of additional software to take advantage of the service.
- furthermore, an important objective is to provide both an homogenized and user-friendly interface to the PNPS service. This requirement is a special challenge as the interface should also provide sophisticated means to describe and handle complex queries, which might consist of subqueries, to various heterogeneous data repositories.
- finally, it is vital for a powerful broker interface to be user-adaptive, to allow user interaction during the request execution and to also dynamically display results. This is especially important as there is often only a thin line between specifying too many or too few attributes to obtain the desired answer. The interface should therefore be able to support flexible, interactive query execution which allows the adjustment of complex requests on the fly.

4.3.2 Approach

In order to guarantee availability and user-friendly access to PNPS the graphical user interface was designed on the basis of a World-Wide Web browser and can therefore be easily accessed by means of widespread browsers like Mosaic, Netscape, or HotJava. In addition, this minimizes the effort to get used to the new broker service.

With respect to the required adaptivity and dynamics of the interface, a realization based on cgi-scripts would have caused problems (each single task has to be performed on the server site even if this is not needed in all cases). Moreover, the hypertext-markup language was developed to construct static web pages. For this reason we decided to design and implement the user interface based on Java-*applets*. The applets can be accessed via an html-page and are directly executed at the client side. The Java-based approach provides a very flexible choice of working environment for the PNPS brokering service. Alternatively, there are other similar languages which might offer the same possibilities in the future, e.g. Python, which can then be integrated into the Grail browser.



Go To: http://la-lure:5556/cgi-bin/cbkb

Specification of:	reqpos2		
Choose result type for next (sub)query specification:	physics		
Select result attribute delivering answer:	author		
title	=	--	optional
author	=	--	optional
keyword	=	energy	obligatory
journal-series	answer of	reqpos3	obligatory
date (from)	=	from (date >...) 01 01 92	obligatory
date (till)	=	till (date <...) 01 03 96	optional
documenttype	is not	thesis	optional
institute	=	--	optional
freetextquery	--		

Select answer attributes to be shown:

title
author
keyword
date

Currently specified query:

Example: find all objects in data repository PHY
where AUTHOR is result of subquery 'reqpos2'(fi

Figure 2: Form-based dialogue for request specifications

The user-friendly request specification is easy to learn and does not require background information on the internally used constraint format, the request syntax of particular data repositories nor on their internal data format. After the user has specified the request (via the form-based dialogues, as illustrated in Fig. 2), the specifications are automatically transformed into a corresponding constraint structure. The generated constraint-based query is then communicated to the generic broker, which further initiates all steps (decomposition of requests into subrequests, threshold checks, recomposition of answers, etc.) to answer the specified request.

Apart from this, the homogenization of the request specification is achieved by presenting a set of possibilities which depend only on the search domain, no matter which of the included data repository and external search tools are contacted to answer the request (e.g. fields for author, title, date, etc., for bibliographical searches). This is achieved by transforming the request into a possibly more general, less precise, external search format of the corresponding external data repository, and by later filtering out the inappropriate results in the constraint solver. In other words, the local constraint solving capabilities are exploited to obtain the results that match against the attributes specified via the user interface. For instance, imagine a query for finding a particular author. Imagine further that the search interface for the backend data repository (which has author attributed information) can only handle flat, i.e. attribute-less keyword searches. Searching for the author name will result in a superset of expected answers, i.e., the search interface will provide the documents where the author name appears as an author as well as where the author name is mentioned somewhere in the full text, for example as a cross-reference. All these “hits” are propagated, but only the expected answers, filtered out through local constraint solving, are presented at the user level.

For these reasons, the user interface includes the following specification functionalities:

- default set of structured search attributes depending on the search domain with supplementary precedence operators.
- a “free-text” entry field which allows additional attributes or untyped keywords as well as combinations of them.
- transitive combination of partial requests viz the values of an attribute. These attributes are required for specifying a request that is fed through extraction from the output fields of the answers to precedent requests. A typical example is the generation of an *author/co-author net*: starting from an author name, the broker system extracts the co-authors from all publications received as answers, and searches for these co-authors in the same recursive way.

  With regard to the fact that various data repositories (e.g. physics, chemistry, computer sciences) can be accessed using a homogeneous interface, the capabilities for request combination can also be used to perform *interdisciplinary* searches. This type of search makes it necessary to run several partial requests on different information servers either consecutively or in parallel, and to combine the partial results in order to build up the final answers. Even in this case, the user only has to provide information about the content-based interdependencies between the partial requests by graphically linking the precedent partial request to the corresponding attribute of the subsequent request.

  A corresponding ’natural-language’-like representation of the specified query is generated during request specification in the input form. This should help the user to control the degree of request combinations.
- threshold specifications which, for instance, allow the user to specify whether a particular field entry is obligatory, before submitting a subsequent partial request.

The form-based request dialogue is quite self-descriptive. Search strings are specified by attribute values and content-based interdependencies between requests, rather than using a formal specification language. In order to enable the sophisticated user to enlarge the default specification set according to his needs, the free-text input fields allow adding additional attributes. Further explanations concerning admissible requests can be obtained by the context-sensitive help service.

After the complete request has been submitted by the user, a check is performed to ensure that sufficient, and only valid attributes, are chosen. If this is not the case, the user is immediately informed, before a time-consuming external request is initiated.

The final results of a single or complex query are displayed via the WWW-based user interface. For this reason, the constraint representation of the answer is retransformed into a more user-friendly attribute/value format which may be presented to the user at different levels of granularity. For each answer given, the maximal available information (which may also include abstracts, matched lines of plain text) can be accessed by a provided hyperlink.

The system displays answers immediately, and not when all external searches have been completed. This is important to be able to react quickly, and, if needed, to change the search specification, i.e., the user should be able to see first results as soon as possible, in order to give him/her an impression whether the request might be successful or should be modified.

This interaction, together with the presentation mode, should facilitate and lead to a symbiosis between tool-based search and interactive browsing. With the aid of Java, it is also possible to support and facilitate further propagation of the presented results, e.g. removing, grouping, or saving selected items.

4.4 Backends

The backends of the knowledge broker system represent the intermediate level between the broker capabilities and the various document archives. It consists of a mediator and a set of data repository dependent interfaces. The communication between the mediator and these archive-interfaces is realized as a constraint exchange protocol.

Each of the various database interfaces encapsulates the information and capabilities which are necessary

- to generate the corresponding external search request from the constraint-based query specification.
- to submit the requests to the archive management software with the help of a suitable protocol (e.g. http).
- to parse the received answers.
- to retranslate them into a set of attribute/value pairs representing the information of a single search hit.

Each “hit” is then extended to a valid constraint-based answer and returned to the knowledge brokers via the mediator. Using available attribute tags or format information, the attribute names are derived from the answers returned by the external data repository.

For example, the answer to a search for documents matched by the attribute ’keyword: energy’ would result in additional attribute/value pairs like:

TITLE: On radio detection of ultra-high energy neutrinos in antarctic ice
AUTHOR: George M. Frichter
AUTHOR2: John P. Ralston
AUTHOR3: Douglas W. McKay
SUBMISSIONDATE: 20-07-1995

REVISIONDATE: 21-07-1995
REPORTNR: KITCS95-1-3
EMAIL: frichter@poincare.math.ukans.edu
PAPER: gopher://physinfo.uni-augsburg.de:70/00/archiv/astro-ph/9507/[...]
ABSTRACT: Interactions [...] energies.

We are currently focusing primarily on the integration of external archives accessible via the http protocol. However, the architecture is flexible enough to incorporate archives which are accessed via a Z39.50 interface or SQL-queries.

For test purposes the following four archives are used (each of these archives is stored on non-commercial servers):

1. the PhysDoc server at the University of Oldenburg,
2. the PhysDiss server at the University of Oldenburg,
3. the PhysInfo server at the University of Augsburg,
4. the collection of computer science bibliographies at the University of Karlsruhe.

The major problems which occur when integrating new data repositories arise from the heterogeneity of the answer format returned by the data repositories. Most often, the answers suffer from not providing a sufficient amount of significant, structured document information. This makes it hard to apply the information found to initiate subsequent requests, or to perform conformity checks on the retrieved results.

5 Conclusion

We have applied agent technology to the practical problem of information gathering in the context of a network publication system. The approach that we have illustrated provides an agent-oriented integration of different technologies such as interactive graphical user interfaces, remote programming tools like Java, and constraint solving. We have shown the advantages of this approach from the point of view of handling partial information, the combination of knowledge from multiple heterogeneous sources, and user-friendliness.

Public access to the system thus developed will be available in June 1996.

References

Andreoli, J.-M., Borghoff, U. M., Pareschi, R. (1994). Constraint-Based Knowledge Brokers. In Hong, H., editor, *Proc. 1st Int. Symp. on Parallel Symbolic Computation (PASCO '94)*, pp. 1–11, Hagenberg/Linz, Austria, September. Lecture Notes Series in Computing **5**, Singapore, New Jersey, London, Hong Kong: World Scientific.

Andreoli, J.-M., Borghoff, U. M., Pareschi, R., Schlichter, J. H. (1995). Constraint Agents for the Information Age. *J. Universal Computer Science*, **1**:12, 762–789. Electronic version available at http://hyperg.iicm.tu-graz.ac.at/Cjucs_root.

Andreoli, J.-M., Borghoff, U. M., Pareschi, R. (1996). The Constraint-Based Knowledge Broker Model: Semantics, Implementation and Analysis. *J. Symbolic Computation*.

Arcelli, F., Borghoff, U. M., Formato, F., Pareschi, R. (1995). Tuning Constraint-Based Communication in Distributed Problem Solving. In *Proc. 1st Int. Workshop on Concurrent Constraint Programming (CCP '95)*, Venice, Italy, May.

Barbara, D., Clifton, C. (1992). Information Brokers: Sharing Knowledge in a Heterogeneous Distributed System. Technical Report MITL–TR–31–92, Matsushita Information Technology Lab., Princeton, NJ, October.

Barbara, D. (1993). Extending the Scope of Database Systems. Technical Report MITL–TR–44–93, Matsushita Information Technology Lab., Princeton, NJ.

Borghoff, U. M., Schlichter, J. H. (1995). On Combining the Knowledge of Heterogeneous Information Repositories. Technical Report, Rank Xerox Research Centre, Grenoble Lab., France, September.

Borghoff, U. M., Pareschi, R., Arcelli, F., Formato, F. (to appear). Constraint-Based Protocols for Distributed Problem Solving. *Science of Computer Programming*.

Bowman, C. M., Danzig, P. B., Hardy, D. R., Manber, U., Schwartz, M. F. (1994). The Harvest Information Discovery and Access System. In *Proc. 2nd Int. World-Wide Web Conf.*, pp. 763–771, Chicago, IL, October.

Bowman, C. M., Dharap, C., Baruah, M., Camargo, B., Potti, S. (1994). A File System for Information Management. In *Proc. Conf. on Intelligent Information Management Systems*, Washington, DC, June.

CACM, **37**:7 (1994). Special Issue on Intelligent Agents. *Communications of the ACM*.

Callan, J. P., Croft, W. B., Harding, S. M. (1992). The Inquery Retrieval System. In *Proc. 3rd Int. Conf. on Database and Expert Systems Applications*, pp. 78–83, September.

Callan, J. P., Lu, Z., Croft, W. B. (1995). Searching Distributed Collections with Inference Networks. In *Proc. 18th ACM SIGIR Conf. on Research and Development in Information Retrieval*, Seattle, WA, July.

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., Widom, J. (1994). The Tsimmis Project: Integration of Heterogeneous Information Sources. In *Proc. IPSJ Conf.*, Tokyo, Japan, October. Los Alamitos, CA: IEEE Comp. Soc. Press.

Emtage, A., Deutsch, P. (1992). Archie: An Electronic Directory Service for the Internet. In *Proc. Usenix Conf. Winter ’92*, pp. 93–110, Sunset Beach, CA, January. Berkeley, CA: Usenix Association.

Fikes, R., Engelmore, R., Farquhar, A., Pratt, W. (1995). Network-Based Information Brokers. In *Proc. AAAI Spring Symp. Series on Information Gathering from Distributed Heterogeneous Environments*, Stanford, CA, March.

Freeman-Benson, B. N. (1990). Kaleidoscope: Mixing Objects, Constraints, and Imperative Programming. In *Proc. OOPSLA/ECOOP ’90*, pp. 77–88, Ottawa, Canada, October. ACM SIGPLAN Notices **25**:10.

Gravano, L., Garcia-Molina, H. (1995). Generalizing Gloss to Vector-Space Databases and Broker Hierarchies. In Dayal, U., Gray, P. M. D., Nishio, S., editors, *Proc. 21st Int. Conf. on Very Large Data Bases*, pp. 78–89, Zurich, Switzerland, September. San Francisco, CA: Morgan Kaufmann.

Grötschel, M., Lügger, J. (1995). Aufbau elektronischer Informations- und Kommunikationsstrukturen. Technical Report TR 95-10, Konrad-Zuse Zentrum für Informationstechnik Berlin, September.

Hentenryck, P. v., Saraswat, V. (1995), editors. *Principles and Practice of Constraint*

Programming, Cambridge, MA: MIT Press.

Henz, M., Smolka, G., Würtz, J. (1995). Object-Oriented Concurrent Constraint Programming in Oz. In Hentenryck, P. v., Saraswat, V., editors, *Principles and Practice of Constraint Programming*, pp. 27–48. Cambridge, MA: MIT Press.

Kahle, B., Medlar, A. (1991). An Information System for Corporate Users: Wide Area Information Servers. *Connexions: The Interoperability Report*, **5**:11, 2–9.

Manber, U., Wu, S. (1994). Glimpse: A Tool to Search Trough Entire File Systems. In *Proc. Usenix Conf. Winter '94*, pp. 23–32, San Francisco, CA, January. Berkeley, CA: Usenix Association.

Obraczka, K., Danzig, P. B., Li, S.-H. (1993). Internet Resource Discovery Services. *IEEE Computer*, **26**:9, 8–22.

Sanchez, J.A. (1994). User Agents in the Interface to Digital Libraries. In *Proc. Conf. on Digital Libraries*, pp. 217–218, College Station, TX, June.

Saraswat, V. A. (1989). *Concurrent Constraint Programming Languages*. PhD thesis, Carnegie-Mellon University, Pittsburg, PA.

Schwartz, M. F., Emtage, A., Kahle, B., Neuman, B. C. (1992). A Comparison of Internet Resource Discovery Approaches. *Computing Systems*, **5**:4, 461–493.

Wooldridge, M., Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, **10**:2, 115–152.